**List/Dict comprehensions, Generator expressions**

**1. List comprehension (списковое включение)** – выражение, позвояющее сгенерировать новый объект списка на основании уже имеющейся последовательности данных в одну строку **sequence = range(1, 11) # последовательность целых чисел от 1 до 10**

# генерация нового списка, используя обычный цикл for и добавление нового элемента в список

**result = [] # создаем пустой список, который нужно будет заполнить**

**for x in sequence: # бежим циклом по нашей последовательности  
 result.append(x) # добавляем элемент последовательности в конец списка**Аналог с использованием **List comprehension:  
  
result = [x for x in sequence]***Поддерживают фильтрацию по условию (Постфиксная запись условия):* **result = [x for x in sequence if x % 2 == 0]**# идет после **for**, и **else** написать в постфиксном условии НЕЛЬЗЯ!# в результат попадут только четные числа:   
# [2, 4, 6, 8, 10]  
*Префиксная запись условия: поддерживается тернарный оператор* ***if/else*****result = [x if x % 2 == 0 else 0 for x in sequence]**# Нечетные числа будут заменены на **0**, четные останутся как есть:  
# [0, 2, 0, 4, 0, 6, 0, 8, 0, 10]  
  
**ЗАПОМИНАЕМ:**  
- префиксная запись (ДО цикла for) – преобразование элементов из исходной последовательности в зависимости от условия, поддерживает else  
- Постфиксная запись (ПОСЛЕ цикла for) – фильтрация элементов из исходной последовательности в зависимости от условия, НЕ поддерживает else

Пример **list comprehension** в связке c **enumerate:**  
  
**list1 = list(range(1, 11))  
list2 = list(range(11, 21))**  
  
# хотим получить список пар, где первый элемент пары – это элемент списка list1, второй элемент пары – элемент списка list2 (списки равной длины, это ВАЖНО!)

**result = [(element, list2[idx]) for idx, element in enumerate(list1)]**# [(1, 11), (2, 12), (3, 13), (4, 14), (5, 15), (6, 16), (7, 17), (8, 18), (9, 19), (10, 20)]

**2. Dict comprehension (словарное включение)** – выражение, позвояющее сгенерировать новый объект словаря на основании уже имеющейся последовательности данных (или другого словаря) в одну строку  
  
Генерация нового словаря из последовательности целых чисел, где **ключ словаря** – строка ‘TEST’ + число последовательности, а **значение по ключу** – само число последовательности  
  
**new\_dict = {f‘TEST\_{x}’: x for x in range(1, 11)}  
# {'TEST\_1': 1, 'TEST\_2': 2, 'TEST\_3': 3, 'TEST\_4': 4, 'TEST\_5': 5, 'TEST\_6': 6, 'TEST\_7': 7, 'TEST\_8': 8, 'TEST\_9': 9, 'TEST\_10': 10}**